Advances in Physics Theories and Applications
ISSN 2224-719X (Paper) ISSN 2225-0638 (Online)
Vol.50, 2015
www.iiste.org

Comparison of Exact and Numerical Solutions with Special Attention to First Order Ordinary Differential Equations

Kassaye Bewketu Zellelew
Department of Physics, College of Natural and Computational Sciences, Wolaita Sodo University, Wolaita Sodo, Ethiopia.

ABSTRACT

In this paper I solved three first-order ordinary differential equations (ode) both analytically and numerically using 4^{th} order Runge-Kutta method (RK4). I selected differential equations which can also be solved analytically so as to compare the numerical solutions with the analytical solutions and see the accuracy of the 4^{th} order RungeKutta method in solving ordinary differential equations of type linear, separable and exact. Both solutions were obtained by employing a computer program written in FORTRAN 90/95. The absolute errors associated with different step sizes have been calculated and the efficient step size for the three types of odes under consideration has been identified. I found out that this numerical method is computationally more efficient and very accurate in solving first-order ordinary differential equations of the three types. This is verified from the relatively small (negligible) differences between the numerical and analytical values (absolute errors).To illustrate the efficiency of the method and for better visualization of its accuracy, the numerical and analytical solutions were plotted against the independent variable. For the differential equations under consideration, the efficient step size (the one with smallest average absolute error) is h = 0.100. When the step size decreases from 0.500 to 0.100, both the relative and absolute errors show a slight decline but they show a slight rise when the step size decreases further from 0.1 to 0.02. This is due to over accumulation of round off errors. Given step size h = 0.100, 4^{th} order Runge-Kutta method is found to be the most efficient for solving the linear ode. The possible reason for this is the relatively smallest degree (extent of nonlinearity) of the analytic solution associated with the linear ode. Further analysis should be made for detailed reasoning.

Key words: Numerical solution, analytic solution, Runge-Kutta method, efficient step size.

1. INTRODUCTION

An ordinary differential equation (ODE) is an equation involving derivatives of an unknown quantity with respect to a single variable. There are many physics problems that involve first order Odes. For example resistance, inductances, electrical circuits and radioactive decays. Ordinary differential equations also appear in numerous problems in population biology and engineering giving mathematical descriptions of some phenomena. The numerical analysis of differential equations describes the mathematical background for understanding numerical methods giving information on what to expect when using them.

For studying numerical methods as a part of a more general course on differential equations, many of the basic ideas of the numerical analysis of differential equations are tied closely to theoretical behavior associated with the problem being solved.

Differential equations can describe nearly all systems undergoing change and are essential parts of many areas of mathematics, from fluid dynamics to celestial mechanics. They are used by mathematicians, physicists and engineers to help in the designing of everything from bridges to ballistic missiles.
Ordinary Differential Equations (ODEs) are one of the most important and widely used techniques in mathematical modeling. However, not many ODEs have an analytic solution and even if there is one, usually it is extremely difficult to obtain and it is not very practical (S. Amen · P et al., 2004). This leads to the need for numerical integration of the Initial Value Problem (IVP) for odes.
Most efforts to increase the order of Runge-Kutta method have been accomplished by increasing the number of Taylor's series terms used and thus the number of function evaluations. Several authors have considered various approximations to reduce the number of stages and the storage requirements of high-order Runge–Kutta methods. However, not so much has been researched about the relationship between a step size and magnitude of absolute error and also the determination of efficient step size corresponding to a given ordinary differential equation.

2. METHODS

Three ordinary differential equations (one each from first order linear, separable and exact) have been selected purposely. The selection was carefully done so that the differential equations can be solved both analytically and

numerically so as to compare the solutions and see the accuracy of the 4th order Runge-Kutta method. Moreover, the interval of validity for the analytical solutions are carefully planned to include the interval [1, 11] from which the value of the independent variable x is allowed to vary while generating the exact and numerical solutions by running the FORTRAN 90/95 code. Based on this selection, the IVPs for ordinary differential equations $x\frac{dy}{dx} + 2y = x^2 - x + 1$, $y(1) = 0.5$ from linear type, $\frac{dy}{dx} = \frac{3x^2+4x-4}{2y-4}, y(1) = 3$ from separable type and $2xy - 9x^2 + (2y + x^2 + 1)\frac{dy}{dx} = 0, y(0) = -3$ from exact were selected with the intention indicated above.

The numerical and analytical solutions were obtained by varying the values of x in the interval [1,11]. This was repeatedly done for each step size (h = 0.500, 0.250, 0.200, 0.100, 0.05, 0.025 and 0.020). The comparison between the results was done by graphical means and by calculating the difference between analytical and numerical values and putting in them tabular form. The results obtained analytically and numerically were plotted against the x-values using excel 2010.

3. RESULTS and DISCUSSIONS

By employing, 4th order RK method, numerical and analytical solutions have been determined for different step sizes and different value of x in the interval [1, 11]. These solutions were then compared with each other for different x values in the same interval. The computer algorithms (FORTRAN 90/95 codes) which are shown in appendices A, B and C were used to generate the values related to the analytic and numerical (using RK4) solutions. As it can be observed from figures 1, 2 and 3 and also from tables 4, 5 and 6 both solutions are approximately the same.

3.1 Graphical analysis of the results obtained by numerical and analytical approaches

As can be seen in the figures below (figures 1, 2 and 3) the exact values (indicated by the dotted lines) and the numerical values(indicated by the solid lines) for step size h = 0.100 are approximately the same. Figures 1, 2 and 3 are obtained by plotting the x-values against exact and RK4 values using the excel 2010 spread sheet. We can see that the two graphs overlap indicating the very good accuracy of the method. The accuracy of the method can also be seen from the tables 4, 5 and 6. Similar situation also applies for other step sizes although not indicated here. This shows that 4th order Runge-Kutta methods are very powerful to solve an ode which has complicated analytic solution.

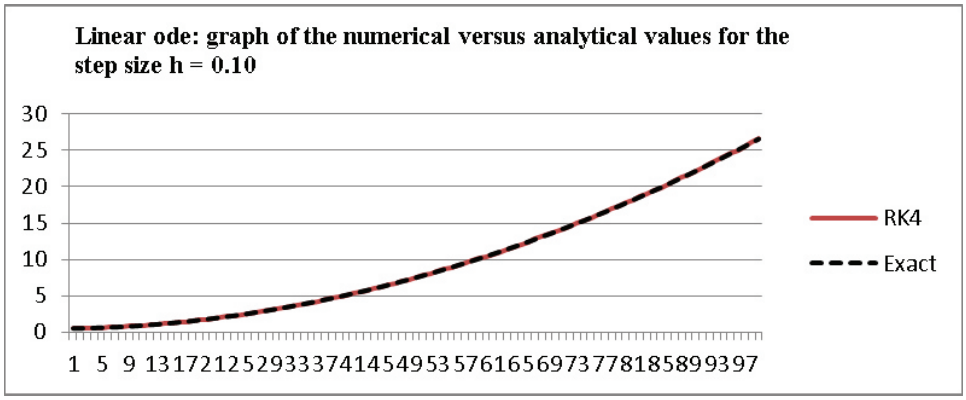

Figure 1

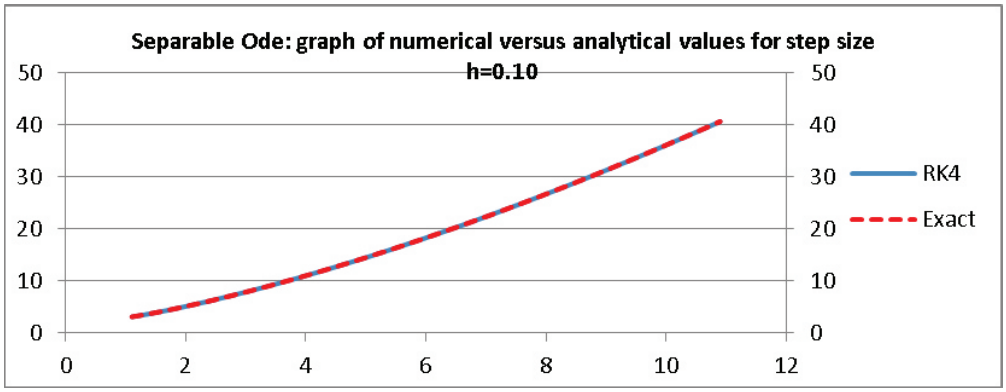

Figure 2

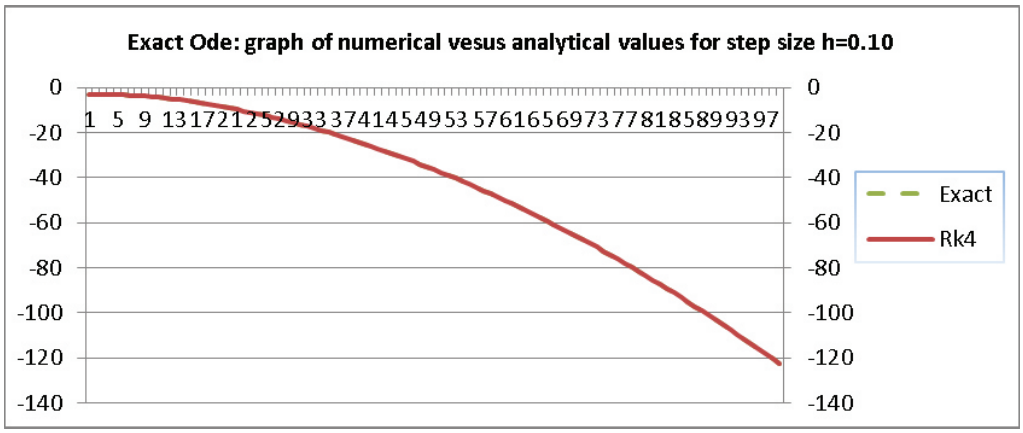

Figure 3

3.2 Tabular representation of the results obtained by numerical and analytical approaches

Tables 1, 2 and 3 are obtained by applying the computer algorithms indicated in appendices A, B and C. As we can see from the tables 1, 2 and 3, the exact values and the numerical values are approximately equal to each other. The average relative errors in percent for the step sizes h= 0.5, 0.25, 0.2, 0.1, 0.05, 0.025 and 0.02 are respectively $9.2206X10^{-3}$, $5.7334X10^{-4}$, $9.126x10^{-5}$, $5.7631x10^{-5}$, $1.1692X10^{-4}$, $1.8749X10^{-4}$ and $1.8750X10^{-4}$ for the linear ode; While the average absolute errors for the same step sizes are respectively $1.3314x10^{-4}$, $8.2345x10^{-6}$, $6.5424x10^{-6}$, $5.6823x10^{-6}$, $1.5706X10^{-5}$, $1.1320X10^{-5}$ and $3.5915X10^{-5}$. The average relative and absolute errors are also relatively small for the separable and exact Odes. This indicates how accurate the 4^{th} order Runge Kutta methods are for numerically solving first order ODE of different types.

Sample tables showing x-values, values obtained from 4^{th} order Runge-Kutta method (RK4), analytical (Exact) values, percent relative errors (Er%) and absolute errors (Ea) for step size h= 0.500 are given below. Tables corresponding to the remaining step sizes are not indicated here due to limited space.

Table 1: Linear ODE: Numerical and analytical values for the step size h = 0.500

x	RK4	Exact	Er (%)	Ea
1.5	0.60000002	0.59953701	7.72277415E-02	4.63008881E-04
2	0.85454929	0.85416663	4.47994359E-02	3.82661819E-04
2.5	1.2427989	1.2424999	2.40625273E-02	2.98976898E-04
3	1.7594962	1.7592592	1.34709012E-02	2.36988068E-04
3.5	2.402828	2.4026361	7.98818283E-03	1.91926956E-04
4	3.1720335	3.171875	4.99856891E-03	1.58548355E-04
4.5	4.0667486	4.0666151	3.28318262E-03	1.33514404E-04
5	5.0867805	5.0866666	2.24044733E-03	1.13964081E-04
5.5	6.2320199	6.2319212	1.58386619E-03	9.87052917E-05
6	7.5024009	7.5023146	1.15041202E-03	8.63075256E-05
6.5	8.8978815	8.8978052	8.57446808E-04	7.62939453E-05
7	10.418435	10.418367	6.49918278E-04	6.77108765E-05
7.5	12.064042	12.063981	5.05928823E-04	6.10351563E-05
8	13.83469	13.834636	3.92922782E-04	5.43594360E-05
8.5	15.73037	15.73032	3.15257814E-04	4.95910645E-05
9	17.751074	17.751028	2.57880078E-04	4.57763672E-05
9.5	19.896797	19.896757	2.01310802E-04	4.00543213E-05
10	22.167538	22.1675	1.72085143E-04	3.81469727E-05
10.5	24.56329	24.563255	1.39770869E-04	3.43322754E-05
11	27.084053	27.084023	1.12677422E-04	3.05175781E-05

Table 2: Separable ODE: Numerical and analytical values for the step size h = 0.500

x-value	RK4	Exact	E_r(%)	Ea
1.5	3.9703138	3.968502	4.57E-02	1.81E-03
2	5.163507	5.1622777	2.38E-02	1.23E-03
2.5	6.4869819	6.4860897	1.38E-02	8.92E-04
3	7.9167676	7.91608	8.69E-03	6.88E-04
3.5	9.4419909	9.4414377	5.86E-03	5.53E-04
4	11.055843	11.055386	4.14E-03	4.58E-04
4.5	12.753295	12.752907	3.04E-03	3.88E-04
5	14.530299	14.529964	2.30E-03	3.35E-04
5.5	16.383444	16.38315	1.79E-03	2.94E-04
6	18.309767	18.309507	1.42E-03	2.59E-04
6.5	20.306652	20.306419	1.15E-03	2.33E-04
7	22.371758	22.37155	9.29E-04	2.08E-04
7.5	24.502968	24.502777	7.78E-04	1.91E-04
8	26.698351	26.698177	6.50E-04	1.74E-04
8.5	28.956141	28.955982	5.47E-04	1.58E-04
9	31.274708	31.274563	4.64E-04	1.45E-04
9.5	33.652542	33.652409	3.97E-04	1 34E-04
10	36.088245	36.08812	3.49E-04	1.26E-04
10.5	38.580505	38.580391	2.97E-04	1.14E-04
11	41.128101	41.127995	2.60E-04	1.07E-04

Table 3: Exact ODE: Numerical and analytical values for the step size h = 0.500

x-values	RK4	Exact	Er(%)	Ea
0.5	-3.2261953	-3.2260816	3.53E-03	1.14E-04
1	-4.1632023	-4.1622777	2.22E-02	9.25E-04
1.5	-5.9586821	-5.9569306	2.94E-02	1.75E-03
2	-8.5226307	-8.5207977	2.15E-02	1.83E-03
2.5	-11.751644	-11.75	1.40E-02	1.64E-03
3	-15.584427	-15.583005	9.12E-03	1.42E-03
3.5	-19.987198	-19.985973	6.13E-03	1.22E-03
4	-24.940342	-24.939281	4.25E-03	1.06E-03
4.5	-30.431623	-30.430696	3.05E-03	9.27E-04
5	-36.452896	-36.45208	2.24E-03	8.16E-04
5.5	-42.998444	-42.997715	1.69E-03	7.29E-04
6	-50.064079	-50.063427	1.30E-03	6.52E-04
6.5	-57.646629	-57.646046	1.01E-03	5.84E-04
7	-65.743629	-65.743095	8.12E-04	5.34E-04
7.5	-74.353111	-74.352623	6.57E-04	4.88E-04
8	-83.47348	-83.47303	5.39E-04	4.50E-04
8.5	-93.103424	-93.103012	4.43E-04	4.12E-04
9	-103.24185	-103.24146	3.77E-04	3.89E-04
9.5	-113.88783	-113.88747	3.15E-04	3.59E-04
10	-125.04059	-125.04026	2.62E-04	3.28E-04

3.3 Representing the accuracy of 4th order Runge-Kutta method in terms of average relative and absolute errors.

As shown below in table 4, 5 and 6 the exact and numerical value agree to a very high extent with the indicated average relative and absolute errors. The average relative error in percent and the average absolute errors are minimum at the step size h of 0.100 indicating that this step size is the most appropriate one for the problem under consideration. The average relative and absolute errors decrease with decreasing step size from h = 0.5 to h = 0.1. The average relative and absolute errors increase with decreasing step size from h = 0.1 to h = 0.02. The reason for this is over accumulation of round off and truncation errors that in turn led to increased average relative and absolute errors. This shows that decreasing the step size decreases the error of a numerical method only up to a certain limit depending on the nature of the differential equation under consideration.

Moreover, given h = 0.10, we can see from tables 4, 5 and 6 that 4th order Runge-Kutta method is associated with relatively smallest absolute error (5.6823×10^{-6}) for the linear ode and relatively largest value (2.13×10^{-5}) for the exact ode. We can see that the value associated with the exact ode is about 3.75 times larger than the value for the linear ode. This is due to the nature (the degree of nonlinearity) of the exact solutions related to the differential equations under consideration.

Table 4: Step size h versus average relative and absolute errors for the linear ode

Value of step size (h)	Average relative error (%)	Average absolute error
0.500	$9.2206X10^{-3}$	$1.3314x10^{-4}$
0.250	$5.7334X10^{-4}$	$8.2345x10^{-6}$
0.200	$9.126x10^{-5}$	$6.5424x10^{-6}$
0.100	$5.7631x10^{-5}$	$5.6823x10^{-6}$
0.050	$1.1692X10^{-4}$	$1.5706X10^{-5}$
0.025	$1.8749X10^{-4}$	$1.1320X10^{-5}$
0.020	$1.8750X10^{-4}$	$3.5915X10^{-5}$

Table 5: Step size h versus average relative and absolute errors for the separable ode

Value of the step size (h)	Average relative error (%)	Average absolute error
0.500	$5.81x10^{-3}$	$4.18x10^{-4}$
0.250	$3.46x10^{-4}$	$2.27x10^{-5}$
0.200	$1.64x10^{-4}$	$1.57x10^{-5}$
0.100	$3.59x10^{-5}$	$7.27x10^{-6}$
0.050	$9.63x10^{-5}$	$2.51x10^{-5}$
0.025	$1.41x10^{-4}$	$3.47x10^{-5}$
0.020	$1.34x10^{-4}$	$4.24x10^{-5}$

Table 6: Step size h versus average relative and absolute errors for the exact ode

Value of the step size (h)	Average relative error (%)	Average absolute error
0.500	$6.14x10^{-3}$	$8.32x10^{-4}$
0.250	$3.6x10^{-4}$	$4.70x10^{-5}$
0.200	$1.70x10^{-4}$	$3.76x10^{-5}$
0.100	$4.47x10^{-5}$	$2.13x10^{-5}$
0.050	$8.73x10^{-5}$	$6.05x10^{-5}$
0.025	$1.32x10^{-4}$	$7.80x10^{-5}$
0.020	$1.02x10^{-4}$	$9.23x10^{-5}$

4. CONCLUSIONS

From the results of this project we can conclude that 4^{th} order Runge Kutta methods have a very good accuracy in numerically solving first order ordinary differential equations. If we have a first order ordinary differential

equation that is too complicated to be solved analytically, we can apply the 4th order Runge Kutta methods to solve the ode numerically with a relatively small absolute and numerical errors. Therefore, numerical solutions to first order ordinary differential equations by using Runge Kutta 4th order method plays a very important role for solving a function which is complicated for analytic solution. From this study (as depicted in tables 4, 5 and 6), one understands that the average relative and absolute errors can increase or decrease whenever the step size h decreases. This increment in errors while step size is decreasing is occurring due to the round of errors. Moreover, the most efficient step size (corresponding to the smallest absolute error) for the three selected differential equations is h= 0.100. Furthermore, the 4th order Runge-Kutta method has the smallest absolute error associated with it for the linear ode with this step size. This is due to the relatively smallest degree (extent of nonlinearity) of the analytic solution associated with the linear ode as compared to the other two odes. Further analysis should be made for detailed reasoning.

AKNOWLEDGEMENTS

I am grateful to the almighty God, the merciful and compassionate that helped me in all endeavors of my life. I would like to forward the most heartfelt thanks and appreciations to my family members for their motivations and love.

REFERENCES

B.V Ramana (2008). Higher Engineering Mathematics. JNTU College of Engineering. Tata McGraw-Hill publishing Company limited, New Delhi, 3rd edition.

David Houcque and Robert R. McCormick (2005). Applications of MATLAB: Ordinary Differential Equations (ODE). School of Engineering and Applied Science - Northwestern University.

David Levermore (2012). FIRST-ORDER ORDINARY DIFFERENTIAL EQUATIONS III: Numerical and More Analytic Methods. Department of Mathematics University of Maryland.

Geeken,D, and JOHNSON O.(.2000). Runge- Kutta with higher order derivative approximations, Applied Numerical Mathematics volume **34**; 207-218

Islam, Md.A. (2015) A Comparative Study on Numerical Solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE) with Euler and Runge Kutta Methods. American Journal of Computational Mathematics, volume **5**, 393-404.

Moses A. Akanbi (2010). A THIRD ORDER EULER METHOD FOR NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS. Department of Mathematics, Lagos State University, Lagos, Nigeria. VOL. 5, NO. 8

S. Amen · P. Bilokon · A. Brinley Codd · M. Fofaria · T. Shah(2004). Numerical Solutions of Differential Equations. Imperial college, London

Steven C.Chapra(2002). Numerical Methods for Engineers with software and programming applications. University of Michigan, Department of civil engineering. McGraw-Hill Companies, New York, 4th edition.

APPENDICES

Appendix A:- The computer algorithm that calculates the exact and numerical values of the linear ode(IVP): $x\frac{dy}{dx} + 2y = x^1 - x + 1$, y(1)=0.5 for different step sizes h = 0.5, 0.25, 0.2, 0.1, 0.05, 0.025 and 0.02.

```
program rk4_linear
 implicit none
integer:: xi=1,xf=11, iter, I     ! Variable Declaration and Initialization
real: :h,x,y,exact,er,ea,k1,k2,k3,k4,slope  !Variable Declaration and  Initialization
print*, 'enter the value of step size h'   ! Display or Output statement
read*,h   !Input variables
          x=1.0    ! Independent variable initialization
          y=0.5    ! dependent variable initialization
iter=(xf-xi)/h    !  calculation of number of iteration
print*,'x',','      ','rk4',','    ','exact',','      ','er(%)',','     ','ea' ! Display or Output statement
do i=1,iter   ! looping
! calculations of slopes at different points in the interval
          k1=x+ (1.0/x)-2*y*(1.0/x)-1.0
          k2=(x+0.5*h) + (1.0)/(x+0.5*h)-2*(y+0.5*k1*h)*(1.0)/(x+0.5*h)-1.0
          k3=(x+0.5*h) + (1.0)/(x+0.5*h)-2*(y+0.5*k2*h)*(1.0)/(x+0.5*h)-1.0
          k4=(x+h) + (1.0)/(x+h)-2*(y+k3*h)*(1.0)/(x+h)-1.0
          slope= (k1+2.0*k2+2.0*k3+k4)/6.0      ! average slope
          x=x+h              ! numerical value of independent variable
          y=y+slope*h  ! numerical value of dependent variable
  exact=(1.0/4.0)*x**2-(1.0/3.0)*x+(1.0)/(12.0*x**2)+0.5 ! exact /analytic/ value of the dependent variable
er=abs(((y-exact)/exact)*100.0)
ea=abs(y-exact)
print*, x, y, exact,er,ea
end do ! end of looping
end program rk4_linear  ! end of the main program
```


Appendix B :- The computer algorithm that calculates the exact and numerical values for the separable ode(IVP): $\frac{dy}{dx} = \frac{3x^2+4x-4}{2y-4}$, y(1)=3 for different step sizes h = 0.5, 0.25, 0.2, 0.1, 0.05, 0.025 and 0.02.

```
program rk4_separable
 implicit none
integer:: xi=1,xf=11, iter, I     ! Variable Declaration and Initialization
real::h,x,y,exact,er,ea,k1,k2,k3,k4,slope  !Variable Declaration and  Initialization
print*, 'enter the value of step size h'   ! Display or Output statement
read*,h   !Input variables
          x=1.0    ! Independent variable initialization
          y=3.0    ! dependent variable initialization
iter=(xf-xi)/h    !  calculation of iteration
print*,'x','      ','rk4','    ','exact','      ','er(%)','     ','ea' ! Display or Output statement
do i=1,iter   ! looping
! calculations of slopes
          k1=(3.0*x*x + 4.0*x - 4.0) /(2.0*y-4.0)
          k2=(3.0*(x+0.5*h)*(x+0.5*h) + 4.0*(x+0.5*h) - 4.0) /(2.0*(y+k1*0.5*h)-4.0)
          k3=(3.0*(x+0.5*h)*(x+0.5*h) + 4.0*(x+0.5*h) - 4.0) /(2.0*(y+k2*0.5*h)-4.0)
          k4=(3.0*(x+h)*(x+h) + 4.0*(x+h) - 4.0) /(2.0*(y+k3*h)-4.0)
          slope= (k1+2.0*k2+2.0*k3+k4)/6.0      ! average slope
          x=x+h              ! numerical value of independent variable
          y=y+slope*h  ! numerical value of dependent variable
  exact=2+sqrt(x*x*x+2.0*x*x-4.0*x+2.0)
er=abs(((y-exact)/exact)*100.0)
ea=abs(y-exact)
print*, x, y, exact, er, ea
end do ! end of looping
end program rk4_separable  ! end of the main program
```


Appendix C:- The computer algorithm that calculates the exact and numerical values for the exact ode(IVP): $\frac{dy}{dx}(2y + x^1 + 1) + 2xy - 9x^1 = 0$, y(0)= -3 for different step sizes h = 0.5, 0.25, 0.2, 0.1, 0.05, 0.025 and 0.02.

```
program rk4_exact
 implicit none
integer:: xi=1,xf=11, iter, I     ! Variable Declaration and Initialization
real::h,x,y,exact,er,ea,k1,k2,k3,k4,slope  !Variable Declaration and  Initialization
print*, 'enter the value of step size h'   ! Display or Output statement
read*,h   !Input variables
          x=0.0    ! Independent variable initialization
          y=-3.0    ! dependent variable initialization
iter=(xf-xi)/h    !  calculation of iteration
print*,'x',','      ','rk4',','    ','exact',','      ','er(%)',','     ','ea' ! Display or Output statement
do i=1,iter   ! looping
! calculations of slopes
          k1=(9.0*x*x-2.0*x*y)/(2.0*y+x*x+1.0)
          k2=(9.0*(x+0.5*h)*(x+0.5*h)-
2.0*(x+0.5*h)*(y+k1*0.5*h))/(2.0*(y+k1*0.5*h)+(x+0.5*h)*(x+0.5*h)+1.0)
          k3=(9.0*(x+0.5*h)*(x+0.5*h)-
2.0*(x+0.5*h)*(y+k2*0.5*h))/(2.0*(y+k2*0.5*h)+(x+0.5*h)*(x+0.5*h)+1.0)
          k4=(9.0*(x+h)*(x+h)-2.0*(x+h)*(y+k3*h))/(2.0*(y+k3*h)+(x+h)*(x+h)+1.0)
          slope= (k1+2.0*k2+2.0*k3+k4)/6.0      ! average slope
          x=x+h              ! numerical value of independent variable
          y=y+slope*h  ! numerical value of dependent variable
  exact=(-(x*x+1.0)-sqrt(x**4+12.0*x**3+2.0*x**2+25.00))/(2.0)
er=abs(((y-exact)/exact)*100.0)
ea=abs(y-exact)
print*, x, y, exact,er,ea
end do ! end of looping
end program rk4_exact  ! end of the main program
```